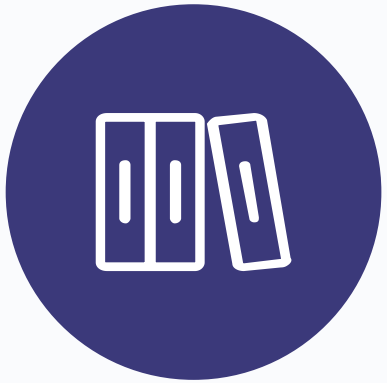

第5章 数据脱敏

成都信息工程大学 白杨 副教授

2024年X月X日

第5章 数据脱敏

本讲内容概要：

01 第一节—敏感数据定义

02 第二节—敏感数据识别

03 第三节—经典脱敏算法

04 第四节—数据匿名化算法

05 第五节—数据合成

第5章 数据脱敏

本讲内容概要：

06 第六节—数据静态脱敏

07 第七节—数据动态脱敏

08 第八节—数据脱敏实例构建

01 1.1 敏感数据定义

敏感数据：是指丢失、滥用、变改或未经许可存取会损害个人隐私或利益、商业秘密，甚至国家的安全和国际关系的数据。敏感数据可理解为特定人群有权知悉的专有数据。

通常敏感数据有以下内容：

- **个人敏感数据**
- **企业敏感数据**
- **国家敏感数据**

01 1.1 敏感数据定义

个人敏感数据

按照我国国家标准《GB/T 35273-2020 信息安全技术个人信息安全规范》中是指一旦泄露、非法提供或滥用可能危害人身和财产安全，极易导致个人名誉、身心健康受到损害或歧视性待遇等的个人信息。

根据《中华人民共和国网络安全法》要求对个人信息和重要数据分开进行评估与定级，再按照就高不就低的原则对数据条目进行整体定级。

从以下三个方面判定是否属于个人敏感信息：

- 泄露：个人信息一旦泄露，将导致个人信息主体及收集、使用个人信息的组织和机构丧失对个人信息的控制能力，造成个人信息扩散范围和用途的不可控。
- 非法提供：某些个人信息仅因在个人信息主体授权同意范围外扩散，即可对个人信息主体权益带来重大风险，应判定为个人敏感信息。
- 滥用：某些个人信息在被超出授权合理界限时使用（如变更处理目的、扩大处理范围等），可能对个人信息主体权益带来重大风险，应判定为个人敏感信息。

01 1.1 敏感数据定义

个人敏感数据

个人敏感信息包括身份证件号码、个人生物识别信息、银行账户、通信记录和内容、财产信息、征信信息、行踪轨迹、住宿信息、健康生理信息、交易信息、14 岁以下（含）儿童的个人信息。

根据《中华人民共和国网络安全法》要求对个人信息和重要数据分开进行评估与定级，再按照就高不就低的原则对数据条目进行整体定级。以个人信息为例，根据个人信息中数据的敏感程度，将数据分级分成四级，一级为低敏感级，二级为较敏感级，三级为敏感级，四级的极敏感级。对业务系统中的数据条目进行拆解，对其中每一个元数据进行分级。

例如：家庭住址为四级，身份证号码为三级，消费账单为二级，然后按就高不就低的原则对数据条目进行整体定级，那么这个**数据条目**应该定为四级。

01 1.1 敏感数据定义

企业敏感数据

从企业或组织角度来看，敏感信息包括客户资料、技术资料、重大决策信息、主要会议纪要、财务预算信息和各种财务报表等高价值数据，这些数据以不同形式存在于企业资产中。

参照 2022 年 9 月 14 日完成的《信息安全技术 网络数据分类分级要求》征求意见稿，行业领域开展数据分类时，应根据行业领域数据管理和使用需求，结合本行业本领域已有的数据分类基础，灵活选择业务属性将数据逐级细化分类。

国家敏感数据

政府数据分级参照 GB/T 31167-2014中，将**非涉密数据分为公开、敏感数据**。

从国家政府部门角度来看，敏感信息是介于保密信息与公开信息之间的特殊信息，这类信息不符合定密标准，不能按照国家秘密的形式进行保护，但是如果公开，却有可能造成某种损害或潜在损害，因此需要限制公开或控制其传播。

第5章 数据脱敏

本讲内容概要：

01 第一节—敏感数据定义

02 第二节—敏感数据识别

03 第三节—经典脱敏算法

04 第四节—数据匿名化算法

05 第五节—数据合成

02 2.1 敏感数据识别

敏感数据识别是为了保护个人隐私和敏感信息免受未经授权的访问和泄露。通过识别敏感数据，组织能够采取相应的安全措施来防止数据泄露和不当使用，符合法规要求，并增强数据安全性。

通常敏感数据识别有以下内容：

- **正则匹配**
- **关键字匹配**
- **基于 NLP 的数据识别**

02 2.1 敏感数据识别

正则匹配

通常情况下，银行卡号、证件号、手机号，**有明确的规则，可以根据正则表达式和算法匹配。**

关键字匹配

姓名、特殊字段，没有明确信息，**可能是任意字符串，可以通过配置关键字来进行匹配。**

基于 NLP 的数据识别

营业执照、地址、图片等，**没有明确规则，可以通过自然语言算法来识别，使用开源算法库。**

02 2.1 敏感数据识别

正则匹配

通常情况下，银行卡号、证件号、手机号，有明确的规则，可以根据正则表达式和算法匹配。

IP地址的正则表达实现如下：

参考代码

```
#精确匹配IP地址
def check_ip(value):
    ip_pattern= r'^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$'
    if re.match(ip_pattern, value):
        print('%s' % s2)
    else:
        print('%s' % s1)
```

手机号正则表达实现如下:

参考代码

```
def check_phone(value):
    phone_pattern=r'^[1](([3][0-9])|([4][5-9])|([5][0-3,5-9])|([6][5,6])|([7][0-8])|([8][0-9])|([9][1,8,9]))[0-9]{8}$'
    if re.match(phone_pattern, value):
        print('%s' % s3)
    else:
        print('%s' % s1)
```

第5章 数据脱敏

本讲内容概要：

01 第一节—敏感数据定义

02 第二节—敏感数据识别

03 第三节—经典脱敏算法

04 第四节—数据匿名化算法

05 第五节—数据合成

03 3.1 经典脱敏算法

数据脱敏的目的是通过一定的方法对敏感数据进行处理以降低敏感数据的敏感程度，或者使得敏感数据不再包含敏感信息内容，从而使敏感数据经脱敏后在保证其可用性、关联性的前提下，达到数据失真的目的。脱敏算法的选择和应用是数据脱敏技术的核心问题。

应根据不同的场景、不同的数据类型、不同的数据特征以及不同的脱敏需求等，选择不同的脱敏算法。

通常经典脱敏算法有以下项：

- **替换**
- **仿真**
- **加密**
- **遮掩**
- **混淆**
- **偏移**
- **均值化**
- **重采样**
- **取整**

03 3.1 经典脱敏算法

替换

替换是指使用具有相似业务特征的伪装数据对原始数据中的敏感数据进行替代，使得原始数据中的相关字段失去原有语义，从而破坏其可读性。为了确保数据的安全性，替换所使用的数据一般具有不可逆性。

替换包括：映射替换、随机替换和参数替换。

仿真

仿真是仿照原始数据中的敏感数据内容生成符合敏感数据原始内容语义和格式的新数据通过相同语义的新数据替换原来的敏感数据.以保证脱敏后的数据能够保持业务数据之间的关联关系，从而使得脱敏后的数据具有较好的可用性。

03 3.1 经典脱敏算法

加密

加密是指通过使用诸如 MD5、Hash、AES 等密码学算法对敏感数据进行加密操作加密处理后的数据与敏感数据的原始内容在逻辑规则和格式上保持一致，外部未经授权的用户只能访问到无实际意义的密文数据，在特定需求场景下，系统也可以给相关需求方提供解密能力以恢复敏感数据的原始内容。

遮掩

遮掩是指通过使用诸如等特殊符号对敏感数据中的部分内容进行掩饰使得敏感数据只选择公开部分原始内容。

该算法在实现数据脱敏、达到保护敏感数据真实信息的同时，较好地保持了敏感数据原始内容的格式，是目前使用较为广泛的脱敏算法。

03 3.1 经典脱敏算法

混淆

混淆是指通过对敏感数据内容在指定条件下进行打乱重排和重新分布，从而破坏与其他字段数据的关联关系，使得混淆后的数据不再具有原始内容的语义。

混淆算法可以保持敏感数据原始内容的组成格式，例如将数字混淆为数字、字母混淆为字母、符号混淆为符号，一般不会影响数据统计特性等业务数据信息。

偏移

偏移主要是通过对敏感数据内容进行随机移位来改变数据内容，偏移算法一般适用于数值型数据。

这种方式通过随机移位改变数字数据，偏移取整在保持了数据的安全性的同时保证了范围的大致真实性，比之前几种方案更接近真实数据，在大数据分析场景中意义比较大。

03 3.1 经典脱敏算法

均值化

均值化一般针对数值型的敏感数据，首先对指定范围的敏感数据进行求和，然后计算出其平均值，最后将脱敏后的数据随机分布在均值附近，以保持数据的总和不发生变化，该算法在一定程度上保证了数据的统计特性。

重采样

从原始数据集中重新采集数据样本，然后将采集到的样本代替原始位置上的数据。这样可以确保脱敏后的数据与原始数据类型相同，同时保护了个人隐私信息。该方法适用于数据分享和数据使用的场景。

取整

将原始数据中的数值进行四舍五入取整，取整的上限以数据的最高位为度量。该方法适用于数据存储和数据使用的场景。

03 3.1 经典脱敏算法-示例

（1）创建原始数据表

在MySQL数据库中创建数据库information，并且在information库中创建表userinfo，表中包含7个字段，分别是first_name、phone、email、web、salary、age、birthday。之后按照表2.1，将数据插入数据表。

first_name	phone	email	web	salary	age	birthday
Aleshia	18357035974	atomkiewicz@hotmail.com	http://www.atomkiewicz.co.uk	16391	21	2000-12-23

03 3.1 经典脱敏算法-示例

（1）创建原始数据表

在MySQL数据库中创建数据库information，并且在information库中创建表userinfo，表中包含7个字段，分别是first_name、phone、email、web、salary、age、birthday。之后按照表2.1，将数据插入数据表。

参考代码
create DATABASE information;

03 3.1 经典脱敏算法-示例

步骤2：创建表userinfo，包含first_name、phone、email、web、salary、age、birthday字段，并且first_name为主键。

参考代码

```
CREATE TABLE userinfo (
    first_name VARCHAR(50) PRIMARY KEY,
    phone VARCHAR(50),
    email VARCHAR(50),
    web VARCHAR(50),
    salary VARCHAR(50),
    age VARCHAR(50),
    birthday VARCHAR(50)
    );
```

03 3.1 经典脱敏算法-示例

步骤3：按照表5.1的内容，将数据插入数据表

参考代码

```
INSERT INTO userinfo (first_name, phone, email, web, salary, age, birthday)
VALUES ('Aleshia', '18357035974', 'atomkiewicz@hotmail.com',
        'http://www.atomkiewicz.co.uk', '16391', '21', '2000-12-23');
```

03 3.1 经典脱敏算法-示例

步骤4：查看表，确认创建成功

参考代码
select * from userinfo;

03 3.1 经典脱敏算法-示例

1)替换

此处以phone字段为例。使用mysql内置函数，生成随机号码，并更新数据表

参考代码

```
UPDATE userinfo SET phone = CONCAT('1', FLOOR(RAND() * (9 - 3 + 1) + 3),
FLOOR(RAND() * (9 - 0 + 1) + 0), FLOOR(RAND() * (9 - 0 + 1) + 0),
FLOOR(RAND() * (9 - 0 + 1) + 0), FLOOR(RAND() * (9 - 0 + 1) + 0),
FLOOR(RAND() * (9 - 0 + 1) + 0), FLOOR(RAND() * (9 - 0 + 1) + 0),
FLOOR(RAND() * (9 - 0 + 1) + 0), FLOOR(RAND() * (9 - 0 + 1) + 0)) where
first_name='Aleshia';
```

03 3.1 经典脱敏算法-示例

2)仿真

此处以first_name字段为例。

步骤1：创建一个名字映射表，该表包含真实英文名和随机生成的英文名

参考代码

```
CREATE TABLE name_mapping (
    real_name VARCHAR(255),
    fake_name VARCHAR(255)
);
```

步骤2：将真实英文名和对应的随机生成的英文名插入到名字映射表中

参考代码

```
INSERT INTO name_mapping (real_name, fake_name)
VALUES
    ('Aleshia', 'Aliice'),
    ('Evan', 'Ella'),
    ('France', 'Fall')
    ;
```

03 3.1 经典脱敏算法-示例

2)仿真

步骤3：更新数据表，将真实的英文名替换为随机生成的英文名

参考代码
UPDATE userinfo JOIN name_mapping ON userinfo.first_name = name_mapping.real_name SET userinfo.first_name = name_mapping.fake_name;

03 3.1 经典脱敏算法-示例

1) 加密

此处以web字段为例。使用MD5加密函数对web字段进行加密，并更新数据表

参考代码
UPDATE userinfo SET web = MD5(web) where first_name='Aleshia';

03 3.1 经典脱敏算法-示例

1) 遮掩

此处以email字段为例。使用LOCATE()函数来找到@符号的位置，然后使用SUBSTRING()函数获取@符号后面的部分，并在CONCAT()中将@符号前面的部分替换为相同数量的星号*。

参考代码
UPDATE userinfo SET email = CONCAT(REPEAT('*', LOCATE('@', email) - 1), SUBSTRING(email, LOCATE('@', email))) where first_name='Aleshia';

03 3.1 经典脱敏算法-示例

1) 混淆

此处以phone字段为例。这个UPDATE语句将会保留手机号码的前三位识别号，然后生成一个7位的随机数字，并使用LPAD()函数来确保生成的数字总长度为8位，然后将生成的随机数字与前三位识别号连接起来。

参考代码
UPDATE userinfo SET phone = CONCAT(LEFT(phone, 3), LPAD(FLOOR(RAND() * 1000000000), 8, '0')) where first_name='Aleshia';

03 3.1 经典脱敏算法-示例

1) 偏移

此处以birthday字段为例。使用DATEDIFF()函数来计算当前日期与birthday字段的日期之间的天数差。然后，使用RAND()函数生成一个0到差值之间的随机整数，表示偏移的天数。最后，使用DATE_ADD()函数将随机偏移的天数加到birthday字段的日期上，并使用LEAST()函数确保偏移后的日期不会超过当前日期。

参考代码

```
UPDATE userinfo SET birthday = LEAST(DATE_ADD(birthday, INTERVAL
FLOOR(RAND() * DATEDIFF(CURRENT_DATE(), birthday)) DAY),
CURRENT_DATE())
where first_name='Aleshia';
```

03 3.1 经典脱敏算法-示例

7)均值化

步骤1：由于需要求平均值，因此这里需要额外插入2条新数据

参考代码

```
INSERT INTO userinfo (first_name, phone, email, web, salary, age, birthday)
VALUES ('Evan', '19378647155', 'zigomalas@gmail.com',
        'http://www.zigomalas.co.uk', '50798', '17', '2004-4-22');

INSERT INTO userinfo (first_name, phone, email, web, salary, age, birthday)
VALUES ('France', '13473682226', 'andrade@hotmail.com',
        'http://www.andrade.co.uk', '22384', '19', '2002-1-21');
```

03 3.1 经典脱敏算法-示例

7)均值化

步骤2：选定一个字段，这里以salary为例。需要创建一个临时表temp_average_salary 来存储平均工资，使用AVG()函数求平均值，再使用ROUND()函数对结果进行四舍五入。

参考代码
CREATE TEMPORARY TABLE temp_average_salary AS SELECT ROUND(AVG(salary)) AS average_salary FROM userinfo;

步骤3：更新salary字段的值，更新值为平均工资加上一个随机偏移量，该偏移量范围在-50到+50之间。

参考代码
UPDATE userinfo SET salary = (SELECT ROUND(average_salary + (RAND() * 100 - 50)) FROM temp_average_salary) WHERE salary IS NOT NULL;

03 3.1 经典脱敏算法-示例

7)均值化

步骤4：删除临时表

参考代码
DROP TEMPORARY TABLE IF EXISTS temp_average_salary;

03 3.1 经典脱敏算法-示例

8)重采样

此处以web字段为例。在子查询中使用ORDER BY RAND()来随机排序web字段，然后通过LIMIT 1选择一个随机的值。接着，在外部查询中，将这个随机选择的值赋给userinfo表中的web字段。

参考代码
UPDATE userinfo SET web = (SELECT web FROM (SELECT web FROM userinfo ORDER BY RAND() LIMIT 1) AS subquery);

03 3.1 经典脱敏算法-示例

9)取整

此处以salary字段为例。代码使用LOG10()函数获取salary的位数，然后通过ROUND()函数四舍五入到最接近的以最高位为度量的整数。例如，对于16391，LOG10(16391)约为4.2142，FLOOR(LOG10(16391))为4，然后-1 * FLOOR(LOG10(16391))为-4。最后，ROUND(16391, -4)就是20000。

参考代码
UPDATE userinfo SET salary = ROUND(salary, -1 * FLOOR(LOG10(salary))) where first_name='Aleshia';

第5章 数据脱敏

本讲内容概要：

01 第一节—敏感数据定义

02 第二节—敏感数据识别

03 第三节—经典脱敏算法

04 第四节—数据匿名化算法

05 第五节—数据合成

04 4.1 数据匿名化算法

为了提高数据集整体的隐私安全性，有效降低数据的敏感程度，实现高可靠的敏感信息保护能力，还存在更为复杂的数据匿名化算法图，包括 K-匿名、T-相近等。

通常，数据匿名化算法有以下项：

- **K-匿名**
- **T-相近**

04 4.1 数据匿名化算法

K-匿名

如果一组公开的数据中，任何一个人的信息都不能和其他至少 k-1 个人区分开，则称该数据满足 K-匿名性，该组数据称为一个等价类。使一组数据满足 K-匿名性的过程称为 **K-匿名化**。

实现 K-匿名性有两个常见的方法：

- 数据抑制：将一些属性的值用星号“ * ”替换。可以取代一列中的所有值或部分值。
- 数据泛化：此种方法将一些属性的精确值用更宽泛的类别取代。

04 4.1 数据匿名化算法

T-相近

T-相近是基于 l-多样性组的匿名化的进一步细化，用于通过降低数据表示的粒度来保护数据集中的隐私。T-相近模型扩展了 l-多样性模型，通过考虑属性数据值的分布，清晰地处理属性值。

T-相近要求每个 K-匿名组中敏感属性值的统计分布与该属性在整个数据集中的总体分布“接近”T-相近将信息获得又分为两部分：关于整体的和关于特定个体的。

T-相近可以防止属性泄露，但不能防止身份泄露。因此，我们有时可能同时需要 K-匿名算法和 T-相近。

第5章 数据脱敏

本讲内容概要：

01 第一节—敏感数据定义

02 第二节—敏感数据识别

03 第三节—经典脱敏算法

04 第四节—数据匿名化算法

05 第五节—数据合成

05 5.1 数据合成

合成数据的运作机制是通过使用不同的技术和方法来生成与真实数据相似但不包含真实个人信息的数据。通过使用合成数据，可以在不暴露真实数据的情况下进行数据分析、模型开发和算法测试。合成数据的生成过程通常涉及使用算法和模型来模拟真实数据的特征和分布，生成的合成数据可以用于替代真实数据进行分析和开发。

以下是一些常见的数据合成方法：

- **生成对抗网络**
- **变分自动编码器（VAE）**
- **数据增强**

05 5.1 数据合成

生成对抗网络

生成对抗网络 GANs 由两个主要组件组成：生成器和判别器。生成器负责生成合成数据，而判别器负责区分真实数据和合成数据。通过不断的对抗训练，生成器和判别器相互竞争和改进，最终生成逼真的合成数据。

变分自动编码器器（VAE）

变分自动编码器是基于原始数据的表示生成新数据的算法。无监督算法学习原始数据的分布，然后使用编码器-解码器架构通过双重变换生成新数据。编码器将输入数据压缩成低维表示形式，解码器根据这种潜在表示形式重建新数据。该模型使用概率计算来实现顺畅的数据重建。

VAE 的缺点也很明显，它是直接计算生成图片和原始图片的均方误差而不是像 GAN 那样去对抗来学习，这就使得生成的图片会有点模糊。

05 5.1 数据合成

数据增强

除了前两种深度学习的方法，还有一些较为基础的合成数据的方法，例如对数据进行翻转、旋转、裁剪、加噪等。

合成数据的使用可以加速开发过程，并降低软件开发生命周期的成本。高质量的合成数据可以显著加快数据科学项目的进展，并提供更多的数据资源。当结合安全的研究环境和联邦学习技术时，合成数据有助于实现数据的去中心化利用。

第5章 数据脱敏

本讲内容概要：

06 第六节—数据静态脱敏

07 第七节—数据动态脱敏

08 第八节—数据脱敏实例构建

06 6.1 数据静态脱敏

依据脱敏操作作用部分以及功能原理的不同，数据脱敏包括静态数据脱敏与动态数据脱敏两种。其中，静态数据脱敏主要应用于测试，开发，培训等非生产环境的场景中，所有的人隐私敏感数据只能在进行脱敏操作之后，进行应用与存储。

通常，数据静态脱敏有以下项：

- **静态数据脱敏简介**
- **静态脱敏的应用场景**
- **静态数据脱敏的价值及意义**

06 6.1 数据静态脱敏

数据静态脱敏简介

静态数据脱敏指使用当前现有计算机技术对数据库中的未处理个人隐私敏感数据进行一系列脱敏处置后，将处置后的个体隐私敏感信息保存到目标数据库之中。

技术：变形、替换、屏蔽、保格式加密（FPE）等算法。

将生产数据导出至目标的存储介质，支持源库脱敏、跨库脱敏、数据库异构脱敏、数据库到文件脱敏、文件到数据库脱敏、文件到文件脱敏。导出后的脱敏数据，实际已经改变了源数据的内容。

静态脱敏应用场景

静态数据脱敏普遍的应用场景主要是在使用信息之前。静态数据脱敏常常应用于需要保护个人隐私或敏感数据的场合，主要应用于以下几个场景：

- 场景一：开发、测试场景
- 场景二：数据分析场景
- 场景三：数据外发场景

06 6.1 数据静态脱敏

静态数据脱敏的价值及意义

价值：静态脱敏产品可以做到对敏感数据进行变形处理的同时，又保证脱敏后数据，不改变数据的类型、格式、含义、分布等使用特征，同时对于保障个人隐私和数据安全具有重要的价值。

意义：静态数据脱敏可以有效地保护个人隐私，避免因信息泄露而导数据静态脱敏产品能够有效防止隐私数据滥用，防止隐私数据在未经脱敏的情况下从企业流出的作用，满足企业既要保护隐私数据，同时又保持监管合规等需求。

第5章 数据脱敏

本讲内容概要：

06 第六节—数据静态脱敏

07 第七节—数据动态脱敏

08 第八节—数据脱敏实例构建

07 7.1 数据动态脱敏

动态数据脱敏是一种数据安全技术，可以在数据库查询和应用程序中动态地隐藏或脱敏敏感数据，以保护数据的隐私和安全。

通常，数据动态脱敏有以下项：

- **动态数据脱敏简介**
- **常用的动态脱敏技术**
- **动态脱敏与静态脱敏的区别**
- **动态脱敏技术应用场景**

07 7.1 数据动态脱敏

动态数据脱敏简介

动态数据脱敏不会改变数据存储的实际值，而是在查询时实时处理数据，以确保只有授权用户能够查看完整的敏感数据，而其他用户只能看到经过脱敏处理的数据。

动态数据脱敏可以应用于各种数据库管理系统，例如 SQL Server、Oracle Database 等，通过配置规则和策略来控制敏感数据的显示和访问权限。它可以对列级别或行级别的数据进行脱敏，支持多种脱敏技术。

07 7.1 数据动态脱敏

常用的动态脱敏技术

数据动态脱敏技术路线主要分为以下三种：

- 结果集解析：

  该技术不改写发给数据库的语句，但需要提前获悉数据表的结构，待数据库返回结果后再根据表结构判断集合内哪些数据需要脱敏，并逐条改写结果数据。

- SQL 语句改写：

  该技术针对包含敏感字段查询的语句进行改写，对于查询中涉及的敏感字段通过外层嵌套函数的方式改写，使得数据库运行查询语句时返回不包含敏感数据的结果集。

- 混合模式脱敏技术：

  由于上述两种技术各有优劣，因此产生了结合上述两种技术的混合模式脱敏技术。例如，在需要查询大量数据时使用基于 SQL 语句改写的技术，在针对少量数据查询或存储过程等情况下，可以使用基于结果集解析的技术。

07 7.1 数据动态脱敏

动态脱敏与静态脱敏的区别

1. **适用场景：**
   - **静态：用于非生产环境，将敏感数据从生产环境中提取并脱敏后，提供给培训、分析、测试、开发等非生产系统使用。**
   - **动态：应用于生产环境，实时判断需要进行脱敏的敏感属性，当低权限用户访问敏感数据时，动态筛选需要脱敏的数据并进行脱敏处理。**
2. 技术路线：
   - 静态：脱敏后的数据以脱敏形式存储于外部介质中，实际上已经改变了存储的数据内容。
   - 动态：存储在生产数据库中的数据并没有发生任何改变。
3. 部署方式：
   - 静态：部署在生产环境与测试、开发、共享环境之间的脱敏服务器上
   - 动态：采用代理部署方式：物理旁路，逻辑串联。

07 7.1 数据动态脱敏

动态脱敏技术应用场景

动态数据脱敏技术需要根据访问对象的权限来选择不同级别的脱敏方式，以保证脱敏后的数据能够满足不同的安全要求。

1. 业务场景：

   动态脱敏系统的主要目的是解决业务系统中普通用户在访问应用系统时对数据权限的控制问题。

2. 运维场景：

   尽管运维人员拥有管理员账号，但业务系统的数据实际上是归业务单位所有，而不是运维部门所有。因此，根据职责分离的原则，需要实现一种技术来允许运维人员访问业务生产数据库，同时也保护敏感数据，避免其被非授权用户查看。

3. 数据交换场景：

   有一种较少见的应用场，两个业务系统之间的数据交换，也称为数据交互。为了保护隐私，交换的数据需要进行脱敏处理。

第5章 数据脱敏

本讲内容概要：

06 第六节—数据静态脱敏

07 第七节—数据动态脱敏

08 第八节—数据脱敏实例构建

08 8.1 数据脱敏实例构建

数据脱敏实例构建步骤包括：识别敏感数据、配置脱敏规则、执行数据脱敏，最后获得脱敏后的数据。

通常，数据据脱敏实例构建有以下项：

- **识别敏感数据**
- **配置脱敏规则**
- **执行数据脱敏**

08 8.1数据脱敏实例构建

识别敏感数据

脱敏工作开始前需要明确数据脱敏目标。基于数据脱敏目标开展敏感数据识别。在识别敏感数据的步骤中，需要识别有哪些敏感数据，以及敏感数据的分类分级都是如何实现的。

主要包含以下几个步骤：

- 步骤一，业务数据的盘点
- 步骤二，敏感特征库的提炼
- 步骤三，数据分类分级的制定

08 8.1数据脱敏实例构建

配置脱敏规则

包括数据脱敏后保持原始特征的分析、数据脱敏算法的选择和数据脱敏算法参数配置：保持原始数据的格式、类型；保持原有数据之间的依存关系；保持引用完整性、统计特性、频率分布、唯一性、稳定性。配置需要脱敏的目标（数据库名/表名/字段名）以及适当的脱敏算法参数，根据业务需求完成其他算法的参数配置。

执行数据脱敏

脱敏操作的实施，根据以上已定义的数据脱敏规则及方法，对不同类型数据进行处理，将数据中的敏感信息进行删除或隐藏。

脱敏数据的验证，在执行完数据脱敏任务后，要检查并验证数据脱敏后的效果，确保所有敏感数据都已被正确处理，同时又能兼顾数据的一致性、完整性和可用性。

谢谢！